CURRENTLY PENDING CLAIMS

Listing of Claims:

1. (Previously Presented) A data processing method comprising:

generating, with a client device, a particular form of a client-resident intermediate user interface (UI) for a server-based and client-side controlled application according to a UI format that is based upon a number of device capabilities for said client device including a client device operating system, a client device maximum receivable packet size, and a list of available client device native UI controls, including the step of supplementing a skeletal UI stored in a first memory location with one or more icons, labels or menu items, or combinations thereof stored in a second memory location, wherein the skeletal UI specifies a layout of the client-resident intermediate UI including respective locations of the one or more icons, labels or menu items, or combinations thereof, and wherein the first memory location and the second memory location are situated on said client device, the skeletal UI and the one or more icons, labels, and menu items being independently updateable from one another;

receiving, at said client device, a number of source data items related to said server-based application; and

populating at least one native UI control used by said intermediate UI with said member of source data items.

2. (Previously Presented) A method according to claim 1, wherein said at least one native UI control is associated with the operating system for said client device.

3. (Previously Presented) A method according to claim 1, further comprising the steps of:

generating an action request in response to a manipulation of said intermediate UI by a user of said client device; and

updating said intermediate UI in response to said action request.

4. (Original) A method according to claim 1, further comprising the steps of:

performing an offline action by said client device while said client device operates in a disconnected mode;

subsequently establishing a session between said client device and a UI server; and

thereafter transmitting, from said client device to said UI server, a command indicative of said offline action.

5. (Original) A method according to claim 1, further comprising the step of saving said number of source data items in a client cache resident at said client device.

6. (Original) A method according to claim 5, further comprising the step of removing client cache items to accommodate said number of source data items.

7. (Original) A method according to claim 6, wherein said removing step selectively removes said client cache items according to a hierarchical preference scheme.

8. (Original) A method according to claim 1, further comprising the steps of:

receiving, at said client device, a client action command related to said server-based application; and

executing said client action command by said client device.

9. (Original) A method according to claim 1, wherein said number of source data items received during said receiving step represent a portion of a larger amount of related data available at a UI server.

10. (Original) A method according to claim 9 wherein:

said larger amount of related data comprises a list of items; and

said number of source data items represents a subset of said list of items.

11. (Original) A method according to claim 9, wherein:

said larger amount of related data comprises a document; and

said number of source data items represents a portion of said document.

12. (Original) A method according to claim 9, wherein:

said larger amount of related data comprises an image; and said number of source data items represents a portion of said image.

13. (Original) A method according to claim 9, wherein:
said larger amount of related data comprises a body of text; and
said number of source data items represents a portion of said body of text.

14. (Previously Presented) A method according to claim 1, further comprising the step of retrieving a command script corresponding to a manipulation of a UI control contained in said intermediate UI, said command script being configured for execution by said client device

15. (Original) A method according to claim 14, further comprising the step of executing, by said client device, said command script in response to the manipulation of said UI control at said client device.

16. (Original) A method according to claim 15, wherein said executing step is performed by said client device in response to an offline manipulation of said UI control at said client device.

17. (Previously Presented) A data processing method comprising:

storing a user interface (UI) form definition locally at a client device, said UI form definition being dictated by a number of device capabilities for said client device including a client device operating system and a client device maximum receivable packet size, wherein the UI form definition includes a list of controls and respective locations of the controls as rendered on the client device, the controls being UI objects provided by the client device operating system or other client-resident application, the UI form definition and the controls being independently updateable from one another;

said client device saving a number of source data items locally, said number of source data items being related to a served-based application;

said client device rendering a UI that is based upon said UI form definition; and

said client device populating said UI with said number of source data items, and wherein said number of source data items comprises a smaller subset than a total number of source data items related to said server-based application, and wherein further subsets of said total number of source data items are downloadable based upon execution of one or more client-side controls.

18. (Original) A method according to claim 17, further comprising the step of receiving, at said client device, said number of source data items from a UI server.

19. (Original) A method according to claim 17, further comprising the steps of:

generating an action request in response to a manipulation of said UI by a user of said client device; and

updating said UI in response to said action request.

20. (Original) A method according to claim 17, further comprising the steps of:

performing an offline action by said client device while said client device operates in a disconnected mode;

subsequently establishing a session between said client device and a UI server; and

thereafter transmitting, from said client device to said UI server, a command indicative of said offline action.

21. (Original) A method according to claim 17, wherein said saving step saves said number of source data items in a client cache resident at said client device.

22. (Original) A method according to claim 21, further comprising the step of removing client cache items to accommodate said number of source data items.

23. (Original) A method according to claim 22, wherein said removing step selectively removes said client cache items according to a hierarchical preference scheme.

24. (Original) A method according to claim 21, further comprising the steps of:

updating said UI in response to a manipulation of a display control rendered by said client device;

requesting an additional number of source data items if said manipulation of said display control triggers a data request command; and

replacing source data items saved in said client cache with said additional number of source data items.

25. (Original) A method according to claim 21, further comprising the steps of: updating said UI in response to a manipulation of a display control rendered by said client device;

retrieving additional source data items from said client cache in response to said manipulation of said display control; and

displaying said additional source data items in said UI.

26. (Original) A method according to claim 17, further comprising the steps of:

receiving, at said client device, a client action command related to said server-based application; and

executing said client action command by said client device.

27. (Original) A method according to claim 17, wherein said UI form definition is dictated by said served-based application.

28. (Previously Presented) A method according to claim 17, wherein at least one of the controls identified by said UI form is a native UI control stored locally at said client device.

29. (Original) A method according to claim 17, wherein said number of source data items saved during said saving step represents a portion of a total number of source data items available via a UI server.

30. (Original) A method according to claim 29, further comprising the steps of:

said client device generating a request for additional source data items; and

said client device receiving, from said UI server, a subsequent portion of said total number of source data items.

31. (Original) A method according to claim 30, wherein said client device generates said request in response to a manipulation of said UI control.

32. (Previously Presented) A data processing method comprising:

obtaining a user interface (UI) form definition for a served-based application, where said UI form definition is based upon a number of device capabilities for a client device including a client device operating system, a client device maximum receivable packet size, and a list of available client device native UI controls;

said client device receiving an instruction to render a particular UI form of a client-resident intermediate UI corresponding to said UI form definition;

said client device rendering said particular UI form with at least one native UI control associated with the operating system for said client device, including supplementing a skeletal UI stored in a first memory location with one or more icons, labels or menu items, or combinations thereof stored in a second memory location, wherein the skeletal UI specifies a layout of the client-resident intermediate UI including respective locations of the one or more icons, labels or menu items, or combinations thereof, and wherein the first memory location and the second memory location are situated on said client device, the skeletal UI and the one or more icons, labels, and menu items being independently updateable from one another;

said client device obtaining a number of data items related to said server-based application; and

said client device displaying said number of data items in said at least one native UI control.

33. (Original) A method according to claim 32, further comprising the step of saving said number of data items in a client cache resident at said client device.

34. (Original) A method according to claim 33, further comprising the step of retrieving said number of data items from said client cache prior to said displaying step.

35. (Original) A method according to claim 32, further comprising the step of requesting said number of data items in response to a manipulation of said at least one native UI control.

36. (Previously Presented) A client device architecture for use with a client device capable of communicating with a data processing server, said client device architecture comprising:

a receive module configured to receive an instruction that identifies a user interface (UI) form definition;

an operating system;

a number of native UI controls provided by said operating system;

a UI form data cache configured to store said UI form definition; and

a UI module configured to generate a particular UI form of a client-resident intermediate UI for a served-based application according to said UI form definition that is based upon a number of device capabilities for said client device including the operating system, a client device maximum receivable packet size, and a list of available client device native UI controls, including the step of supplementing a skeletal UI stored in a first memory location with one or more icons, labels or menu items, or combinations thereof stored in a second memory location, wherein the skeletal UI specifies a layout of the client-resident intermediate UI including respective locations of the one or more icons, labels or menu items, or combinations thereof, and wherein the first memory location and the second memory location are situated on said client device, and to populate at least one of said native UI controls with a number of source data items associated with said server-based application, the skeletal UI and the one or more icons, labels, and menu items being independently updateable from one another.

37. (Original) A client device architecture according to claim 36, further comprising a client cache configured to store said number of source data items.

38. (Original) A client device architecture according to claim 37, further comprising a cache management module configured to remove items stored in said client cache to accommodate said number of source data items.

39. (Original) A client device architecture according to claim 38, wherein said cache management module is further configured to selectively remove said items according to a hierarchical preference scheme.

40. (Previously Presented) A client device architecture according to claim 37, further comprising a cache management module associated with said client cache, wherein:

said UI module is further configured to update said intermediate UI in response to manipulation of a display control rendered in connection with said intermediate UI;

said cache management module is configured to request an additional number of source data items from a remote UI server if said manipulation of said display control triggers a data request command; and

said cache management module is further configured to replace source data items saved in said client cache with said additional number of source data items.

41. (Previously Presented) A client device architecture according to claim 37, further comprising a cache management module associated with said client cache, wherein:

said UI module is further configured to update said intermediate UI in response to manipulation of a display control rendered in connection with said intermediate UI;

said cache management module is configured to retrieve an additional number of source data items from said client cache in response to said manipulation of said display control; and

said intermediate UI module is further configured to display said additional source data items in said intermediate UI.

42. (Original) A client device architecture according to claim 36, wherein said receive module is further configured to receive said number of source data items from a remote UI server.

43. (Original) A client device architecture according to claim 36, wherein said receive module is further configured to receive said UI form definition from a remote UI server.

44. (Original) A client device architecture according to claim 36, wherein said UI form definition is based upon a number of device capabilities for said client device.

45. (Previously Presented) A client device architecture for use with a client device capable of communicating with a data processing server, said client device architecture comprising:

a receive module configured to receive an instruction that identifies a user interface (UI) form definition;

an operating system;

a number of native UI controls provided by said operating system, wherein the UI form definition includes a list of the native UI controls and respective locations of the native UI controls as rendered on the client device, the controls being UI objects provided by said operating system or other client-resident application, the UI form definition and the controls being independently updateable from one another;

a UI form data cache configured to store said UI form definition; and

a UI module configured to generate a UI for a server-based application according to said UI form definition, and to populate at least one of said native UI controls with a number of source data items associated with said server-based application, and

wherein said number of source data items comprises a smaller subset than a total number of source data items related to said server-based application, and wherein further subsets of said total number of source data items are downloadable based upon execution of one or more client-side controls, and wherein said UI form definition is based upon a number of device capabilities for said client device including the operating system and a list of available client device native UI controls.

46. (Previously Presented) A client architecture according to claim 45, further comprising a client cache configured to store said number of source data items.

47. (Previously Presented) A client device architecture according to claim 46, further comprising a cache management module configured to remove items stored in said client cache to accommodate said number of source data items.

48. (Previously Presented) A client device architecture according to claim 47, wherein said cache management module is further configured to selectively remove said items according to a hierarchical preference scheme.

49. (Previously Presented) A client device architecture according to claim 46, further comprising a cache management module associated with said client cache, wherein:

said UI module is further configured to update said UI in response to manipulation of a display control rendered in connection with said UI;

said cache management module is configured to request an additional number of source data items from a remote UI server if said manipulation of said display control triggers a data request command; and

said cache management module is further configured to replace source data items saved in said client cache with said additional number of source data items.

50. (Previously Presented) A client device architecture according to claim 46, further comprising a cache management module associated with said client cache, wherein:

said UI module is further configured to update said UI in response to manipulation of a display control rendered in connection with said UI;

said cache management module is configured to retrieve an additional number of source data items from said client cache in response to said manipulation of said display control; and

said UI module is further configured to display said additional source data items in said UI.

51. (Previously Presented) A client device architecture according to claim 45, wherein said receive module is further configured to receive said number of source data items from a remote UI server.

52. (Previously Presented) A client device architecture according to claim 45, wherein said receive module is further configured to receive said UI form definition from a remote UI server.

53. (Previously Presented) A client device architecture according to claim 45, wherein said UI form definition is based upon a number of device capabilities for said client device further including a client device maximum receivable packet size.